Controlled Experiments in Software Engineering

Florian Fittkau

Christian-Albrechts-University Kiel
Department of Computer Science
24098 Kiel, Germany

Abstract. Today, most processes, methods, and tools in software engineering only rely on experience, lacking empirical evidence for a practical application. For example, an unanswered question for most tools is "Under which circumstances should the tool be used and when not?". For this reason, research should aim to show which technology is useful for whom for which tasks and in which environment [15]. For finding answers to those questions, controlled experiments can be used. This paper describes controlled experiments in software engineering including necessary steps for performing them and related challenges.

1 Introduction

Common wisdom, intuition, speculation, and proofs of concepts are not reliable sources of knowledge [2]. In spite of those sources, experimentation can be a reliable source. However, there is a lacking of experimentation and therefore also validation in software engineering [8]. For most process, tools, and techniques the question "under which circumstances is it better than another?" is unanswered [1].

Experimentation is difficult in software engineering [2]. One problem is the fact that experiments have to be realistic for transfer to industry applications [14]. However, professionals are expensive, the setting have to be as close as possible to an industry setting, and the tasks must be chosen realistically. This paper describes the different steps necessary for performing an experiment in software engineering. Furthermore, special focus will be laid on the validity of experiments in order to provide the ability to assess conducted experiments by others and where to be cautious when conducting own experiments.

Structure of the paper. Section 2 describes foundations of controlled experiments and the different types of experiments. Then, Section 3 illustrates the steps that are needed to perform an experiment. Subsequently, the different dimensions of an experiment, i.e. subject, task, and environment, are described in Section 4. Section 5 stresses the fact that replication is essential for empirical methods. Then, Section 6 defines internal and external validity for an experiment. Section 7 provides an example of a controlled experiment in software engineering. Finally, Section 8 concludes the paper.

2 Controlled Experiments

2.1 Foundations

The key feature common to all experiments is varying something with the intention to discover what happens to something else later. In other words, experiments want to discover the effects that happen due to presumed causes. Shadish et al. [11] define an experiment as “a study in which an intervention is deliberately introduced to observe its effects” . In old definitions of cause and effect, cause and effect depend on each other. Thus, the so-called counterfactual model is needed. It is assumed that David Hume defined the counterfactual model in the 18th-century. In an experiment, we observe what did happen to experimental units, like people or animals, after the treatment, like teaching a new method. The counterfactual is what would have happened to the units, if the treatment was not applied. An effect is then defined as the difference between what happened after the treatment and what would have happened without the treatment. Then, the treatment may be the cause to the effect. However, it might not be the only cause and might not be the cause at all.

Notably, applying the treatment and applying no treatment to the same unit at the *same* time is physically impossible. Therefore, there is a need to approximate the counterfactual by keeping the unit and environment as similar as possible.

In an experiment, the possible cause for an effect is the independent variable that we can vary and the effect is indicated by the dependent variable [4]. For example, an experimenter wants to evaluate whether a new development tool leads to higher productivity in comparison to the old development tool. Then, the independent variable would be usage of the new development tool and the dependent variable would represent productivity measured in lines of code per hour, for instance.

An independent variable is often called a factor or treatment in the experimental design. The different values or classification for a factor are called the levels of the factor. For example, in an experiment that wants to show differences in programming styles between genders, the gender would be a factor, and male and female the levels of the factor.

An important issue is the differentiation between an experiment and a correlational study. Shadish et al. [11] define it as “usually synonymous with nonexperimental or observational study; a study that simply observes the size and direction of a relationship among variables.”

A controlled experiment in software engineering is defined by Sjøberg et al. [15] as “A randomized experiment or a quasi-experiment in which individuals or teams (the experimental units) conduct one or more software engineering tasks for the sake of comparing different populations, processes, methods, techniques, languages, or tools (the treatments).” In other words, controlled experiments are randomized experiments or quasi-experiments. These types are defined in the next section.

2.2 Types of Experiments

Different types of experiments have been developed due to the needs and histories of different sciences. Shadish et al. [11] describe the three types randomized experiment, quasi-experiment, and natural experiment which are described below.

Randomized Experiment A randomized experiment is “an experiment in which units are assigned to receive the treatment or an alternative condition by a random process such as the toss of a coin or a table of random numbers.” [11]

It is assumed that Sir Ronald Fisher popularized this type of experiment. In difference to other types of experiments, the various treatments are assigned to experimental units, e.g. people or animals, by chance in randomized experiments. This method results in probabilistic similar groups on the average. If the assignment is not random, the groups are more likely to differ in special attributes. For instance, one group may have better programmers than another group, if they are assigned by their belonging company. Therefore, the outcome of the experiment can result from this difference between the groups. However, the applied treatment might not be the cause for the effect.

Quasi-Experiment A quasi-experiment is defined as “an experiment in which units are not assigned to conditions randomly.” [11]

Sometimes, random assignment to treatments is not possible. Hence, Campbell and Stanley popularized a design they called quasi-experiments. In this design, the assignment process of subjects to groups is self-selection or administrator-selection. Thus, the groups can differ in many systematic ways because of a non-random selection. Many of those ways can be an alternative explanation of the outcome of the experiment. Wiping out all possible alternative interpretations would be infeasible. Instead, the researcher has to identify plausible alternatives and try to reduce the effects of those differences while designing and performing the experiment.

Natural Experiment A natural experiment is “not really an experiment because the cause usually cannot be manipulated; a study that contrasts a naturally occurring event such as an earthquake with a comparison condition.” [11]

Natural experiments describe a naturally occurring difference between a treatment and a comparison condition. In most cases the treatments are not manipulable like earthquakes. In software engineering those treatments might be defects over time of server machines and the comparison condition might be rising hosting prices, for instance.

3 Steps for Performing a Controlled Experiment

Pfleeger [10] defines six steps for planning and running an experiment. Figure 1 illustrates these steps. They are discussed in Section 3.1 to 3.6.

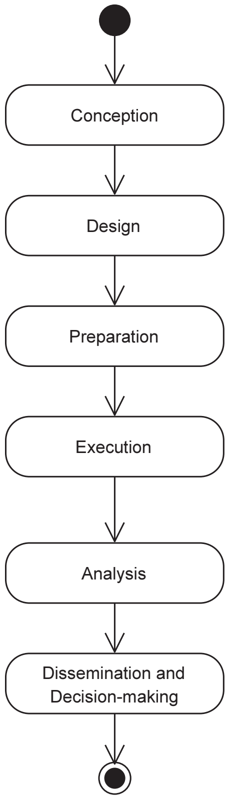

Fig. 1. Steps for performing a controlled experiment defined by Pfleeger [10]

3.1 Conception

The first step includes decisions about the goals and the type of experiment. The type of experiment can be randomized experiment or quasi-experiment. Furthermore, the objective should be clearly formulated in this step so that the objective can be evaluated after the experiment. For this purpose, Pfleeger [10] suggests that it should be stated as a question.

3.2 Design

The design step is the probably most important and complex step in performing an experiment. Therefore, it is split here into translation from an objective into a formal hypothesis and the generation of an experimental plan. Both phases are described below.

Translation from an Objective into a Formal Hypothesis At first, the objective stated in the conception step has to be translated to a formal hypothesis.

There are different types of hypotheses [6]: Universal, existential, and hypothesis about fraction.

An universal hypothesis posits that a circumstance is true for *all* cases. Thus, the hypothesis is falsified, when only one single case exists where the circumstance is not true. The verification of universal hypothesis is impossible, if only a fraction of all cases can be inspected.

An existential hypothesis claims that there exists at least one single case where a determined circumstance is true. The falsification of this type of hypothesis is impossible, if not all cases can be tested. An existential hypothesis is verified, when only one case can be found where the circumstance is true. Notably, not all cases have to be tested.

A hypothesis about fraction proposes a circumstance that holds for a specific fraction of all possible cases. 88% to 95% of all people are right-hand, for instance. Hypothesis about fraction are neither verifiable nor falsifiable, if only a part of all cases can be tested. In this case verification and falsification is impossible because we can only test a subgroup of all cases which may not have exactly the assumed fraction for the specific circumstance. Though, statistical methods can tell us how likely our drawing of our subgroup was.

Often, two hypothesis are made, namely the null hypothesis and the alternative hypothesis. The null hypothesis assumes that there is no difference in the outcome despite the treatment. The alternative hypothesis states that there is a significant difference in the result between the group that received a treatment and the group that received no treatment. For example, the null hypothesis can be that there is no difference in understandability between UML diagrams and box-and-line diagrams. In contrast, the alternative hypothesis would be that there is a higher understandability between UML diagrams and box-and-line diagrams.

This methodology has statistical reasons for later interpretation. We assume that the null hypothesis holds unless the data indicate that this is with high probability, commonly 95 percent, not true.

Generation of an Experimental Plan After defining the hypothesis, an experimental plan has to be generated. The plan provides answers for how to run the experiment.

At this stage the researcher has to define the dependent and independent variables. In conjunction with these variables, he must be conscious about possible extraneous variables. They may lead to wrong results because a extraneous variable, that is not randomized or kept the same across the different groups, could become an influencing factor for the dependent variable, either. Therefore, the results of the experiment may be ambiguous what variable caused the change in the dependent variable, i.e. the independent or extraneous variable. This problem belongs to the class of experimental errors.

There are four major principles, namely replication, randomization, matching, and local control, that reduce experimental errors.

Replication is the repetition of the experiment. By repeating the experiment, the results of the original experiment are confirmed or failed to be confirmed. A replication is important because the results of the experiment might have happened by an extraneous variable that was not controlled. In a replication the not controlled extraneous variable might change and thus not confirm the original results because the results were influenced by the extraneous variable.

Randomization is an important concept to ensure that our result really follows from the treatment. Randomization is the random assignment of subjects to groups. In doing so, the different attributes that vary from subject to subject get assigned to the groups, resulting in a statistical homogeneous distribution of the differences. For example, there are 20 subjects. Ten of them have low programming skills, the other ten have high programming skills. We want to construct two groups. By assigning the programmers by chance, the probability of creating the homogeneous group with five subjects with low programming skills and five subjects with high programming skills is the highest. The major concept about randomization is that we not only make the groups statistical homogeneous in programming skills, by assigning them by chance we also make the groups statistical homogeneous in other attributes like social skill, fatigue, etc.

If randomization is not possible like in a quasi-experiment, the next best method is matching. When using matching the first step is to measure the specific extraneous variable for each subject. After this, the subjects are assigned to the groups so that the means for the measured variable in each group are as similar as possible.

The last principle to reduce the impact of experimental error is local control. Local control reflects how much control we have over the organization of subjects and assignment of treatments. It has two characteristics: blocking and balancing the subjects. Blocking refers to the allocation of subjects to blocks with the goal that the subjects within a block are homogeneous in a special attribute. For example, there might be 30 subjects for testing a new developing method. They come from three different companies. The performance of the different subjects might rely on the company they are working for because there might be different training programs. Therefore, the subjects from the first company are assigned to one block, the subjects from the second company to the second block, and the subjects from the third company to the third block. In each block the ten subjects receive different treatments assigned by chance. Balancing the subjects refers to the concept of blocking and assignment of treatments with the goal to balance the number of subjects assigned to each treatment equally. Balancing can simplify the statistical analysis.

After the generation of the experimental plan, the researcher should know how many units must be tested, from which population they must be drawn, and what experimental factors are involved. Furthermore, he knows which treatment in which condition and which order each subject should receive. He also should know how to measure the dependent and independent variables, i.e. defining the kind of measurement and the measurement scale.

3.3 Preparation

In this step, the preparations for the experiment execution are done. The specific preparation ensues from the experimental plan. Hardware and software must be configured, for instance. Furthermore, the instruction for the experimental units have to be created. A dry run should be done to test the experimental plan and assure that it is complete and the instructions are understandable.

3.4 Execution

The execution steps of the experiment is implied by the experimental plan. Special attention should be laid on the consistent measuring and treatment of the experimental units to assure comparability between the results.

3.5 Analysis

The first part of the analysis step is to review all measurement data. The data has to be complete and useful. The second part of the analysis step is to analyze the sets of data according to statistical principles. For example, the data might be normally or non-normally distributed which must be tested. According to the given distribution and number of samples a test must be chosen like the student's t-test. The result of the statistical analysis is whether the result of the experiment rejects or fails to reject the null hypothesis.

3.6 Dissemination and Decision-making

Everything about the experiment has to be written down such that another person, that is not involved in the experiment, can replicate it. A replication can help to confirm the results.

The results of the experiment may help in three ways. First, they can be used to support decisions about future acting, like which tools or methods will be used. Second, others can use the results to make changes for their development process or environment. Third, other similar experiments with slight variations can be conducted to understand the impact of those controlled changes.

4 Dimensions in a Controlled Experiment

In a controlled experiment are three major dimensions [15]. These dimensions are subject, task, and environment. Section 4.1 to 4.3 describe them.

4.1 Subject

The participants are the possibly most important factor in a controlled experiment. Therefore, the kind of subjects, how they are recruited, and which motivation they have, can have high influence on the results and the generalizability of the experiment.

Sjøberg et al. [13] stress that the choice of the subjects should be as realistic as possible to enable technology transfer from the research community to industry. Therefore, a researcher should be conscious about the target population which he wants to make claims *before* the experiment execution. Only then, he can choose representative subjects for his experiment.

An often criticized choice for subjects are students. They often lack the experience professionals have and hence the experiment may be unrealistic for an industry setting. However, in practice most subjects are students because they can be recruited easily and are much cheaper than professionals. Using professionals in an experiment often requires a high fund of research money.

Another factor is the motivation of the subject. Maybe the subject does not take part free willed because it is forced to. Hence, the subject might conduct the task not seriously or even in a counterproductive way.

4.2 Task

Sjøberg et al. [15] define four general categories for major tasks on software artifacts, namely plan, create, modify, and analyze. In their survey, they report that the most part of experiments in the inspected papers were done in the analysis category. For example, document comprehension tasks are the basis for many software engineering tasks.

Often the duration of tasks is chosen too short [13]. Observable results shall occur in hours in those experiments. In real development, the benefit from a new development tool may show at first after months, for instance. Thus, the short duration of the tasks endangers the generalizability.

4.3 Environment

Controlled experiments in the field of software engineering are often executed in an artificial environment [15, 13]. This concerns the location and the tool environment.

Running an experiment in a typical office environment increases the realism. However, this can lead to threats to the validity of the results because there can be breaks like phone calls or interruptions. In a laboratory environment the generalizability may suffer because it lacks the needed realism.

The tools' setup is another issue why the experiment can lack realism and thereby also the generalizability. If other tools are used in the experiment that the professional is not used to, the results of the experiment may be influenced.

5 Replication

Here, replication refers to an experiment that is run based on the design of a previous experiment. Replication is a crucial instrument for science. Without a first replication the generalizability of results of an experiment are in question.

A naive point of view could argue that replications are not necessary. In fact, identical replication are nearly impossible and not practical. If they were really identical, they would yield exactly the same results. Lindsay and Ehrenberg [9] define two types of replications, i.e. close and differentiated replication. *Close replication* attempts to keep almost all known factors of the original experiment similar. Therefore, a close replication is particularly suitable for first replications to show the generalizability. The term *differentiated replication* refers to the replication of an experiment while changing one known aspect. This type of replication is useful for extending the conditions under which the result still holds.

To enable replications by other researches the experiment has to be described with much detail. There are different problems that prevent this, even if the researcher is very careful. For example, Shull et al. [12] describe the tacit knowledge problem. Tacit knowledge refers to information that is important to the experiment but is not written down in the description because the original research thought it was obvious. This problem can cause a new variability when the researcher, who conducts the replication, applies the treatment in a slightly different way.

For overcoming this and other problems, schemes for describing facts about an experiment are proposed. Höst et al. [5] propose a scheme for describing participants. The scheme classifies participants according to two factors. These factors are incentives and experience of subjects. Incentives refers to the associated motivation of the subjects ranging from an isolated artifact to a project with long-term commitment. The experience of subjects classification ranges from undergraduate student with less than three months recent industrial experience to any person with industrial experience for more than two years.

Jedlitschka and Pfahl [7] provide a proposal of a standardized reporting scheme for an experiment paper. They state that standardized reporting guidelines can yield an easier finding of relevant information, which results in a higher understanding for possible replication. For further details on their proposal, we refer to their paper.

6 Validity

Shadish et al. [11] define four types of validity. The first one is statistical conclusion validity that is the validity about correlation between treatment and outcome. The second is internal validity which is described in Section 6.1. The third is construct validity which is the validity of inferences about the higher order constructs. The last is external validity which is outlined in Section 6.2. Threats to statistical conclusion validity are not described because the statistical foundations are out of the focus of this paper. The construct validity is also not described here because the building of constructs is also not focus of this paper.

6.1 Internal Validity

Internal validity is “the validity of inferences about whether observed co-variation between A (the presumed treatment) and B (the presumed outcome) reflects a causal relationship from A to B as those variables were manipulated or measured” [11]. In other words, an experiment has internal validity, if the changes in the dependent variable (e.g. higher usability) were caused by the manipulation of the independent variable (e.g. the user interface). Shadish et al. define the threats to internal validity that are described below.

Ambiguous Temporal Precedence Ambiguous temporal precedence refers to the issue when it is unclear whether variable A precedes variable B or B happens before A. Then, it can not be concluded which variable is the cause and which is the effect.

Selection Selection is the threat when groups are different in a characteristic that could also influence the dependent variable. This threat often occurs in quasi-experiments where the subjects are not randomly assigned to groups.

History When an event occurs concurrently to the treatment or between two measurements of the dependent variable, the dependent variable can have changed due to the independent or the historical event.

Maturation Subjects can change during an experiment. This includes permanent changes such as physical growth in long-term experiments and temporary changes like fatigue. The concentration of the subjects in an experiment that processes over a few hours changes and this can influence the error rate, for example.

Regression When subjects are selected by their extreme score on the first measurement, the subjects tend to have a score on the second measurement that is more towards the mean. Thus, if an experiment uses extreme score subjects after the first measurement and then applies the treatment, the second measurement may not be the result of the treatment.

Attrition If subjects of one group that are systematically related to the dependent variable drop out, attrition has occurred. For example, the bad programmers drop out of one group, then the group will probably perform better at the end of the experiment. However, this effect may be associated with the higher fraction of programmers with high skill and not the particular treatment like training the group.

Testing If the same test is given to a subject, the subject might remember the answers or knows that it is tested. Then, the subject will probably have a higher score in the second test.

Instrumentation If the instrumentation changes over time, it is also a threat to the internal validity. For example, if the measurement method is observation, the concentration of the observer may change. Therefore, it affects the quality of the measurement and making the results possibly incomparable.

Additive and Interactive Effects of Threats to Internal Validity Threats to validity can combine to produce a bias in an experiment. These threats occur when one group differentiates from another by a threat described above. For example, a selection-maturation effect occurs when the groups mature at different rates.

6.2 External Validity

External validity is the validity of inferences about whether the relationship between cause and effect holds over variations in persons, settings, treatments, or outcomes [11]. In other words, if an experiment has a high external validity, it is possible to generalize the results to other persons, settings, treatments, or outcomes. Shadish et al. define the following possible threats to external validity, i.e. why inferences about the results might not hold over variations.

Interaction of the Causal Relationship with Units If the selection process of the subjects had not been made by chance, the generalizing from this sample to a population might be wrong because the sample is biased. Another type of this threat is the generalizing across populations. This is present when the result holds for different subgroups of people.

Interaction of the Causal Relationship over Treatment Variations The interaction of the causal relationship over treatment variations states that an effect found with one treatment variation might not hold with other variations of that treatment. An important issue in replication is the high probability that the treatment will be altered in some kind.

Interaction of the Causal Relationship with Outcomes If the effect was found by a specific kind of outcome observation, the effect may not hold for other kinds of outcome observations.

Interaction of the Causal Relationship Settings A result that was measured in one setting may not hold in another setting. Therefore, the relationship between dependent and independent variable should be measured across different settings.

Context-Dependent Mediation This threat refers to the situation where there might be a mediating relationship in one context but not in the other. For instance, for some people there might be a direct causal relationship between Scrum and high quality code, but for others there might be no such relationship.

7 Example for a Controlled Experiment

The Sections 7.1 to 7.6 describe an example experiment in software engineering done by Sobel and Clarkson [17] with the steps from Section 3. The steps are only implicitly described in the papers and hence we transfered the given information into the steps.

The experiment is chosen because there exists many information about it in terms of two research papers [16, 17], one comment paper by Berry and Tichy [3], and one response to these comments by the original authors [18].

7.1 Conception

The experiment had the goals to demonstrate the potential of undergraduate students for learning formal analysis techniques, to show the feasibility of teaching formal analysis, and to increase the complex problem solving skills of students. Furthermore, Sobel and Clarkson wanted to give empirical evidence that formal analysis during software development produces "better" programs with regard to the metrics of code correctness, conciseness, and complexity.

The type of analysis is chosen as a quasi-experiment because the choice in which curriculum the students want to take part should not be chosen by chance. Random assignments would mean that the students would take part in a perhaps unwanted three years curriculum, which would be unfair and may influence the results.

Sobel and Clarkson only describe one experiment with much detail for the goal of increasing the problem solving skills of students with the use of formal analysis. Therefore, the following steps refer to this goal.

7.2 Design

There were two groups of students that both took part in an object-oriented design course at the Miami University of Ohio. One group did not learn formal analysis techniques. This group was the control group. The other group studied two semesters with courses of instruction in formal methods before. Hence, they were called the formal methods group.

Translation from an Objective into a Formal Hypothesis The authors stated the alternative hypothesis as that the formal methods group solutions are better than the control group solutions due to their use of formal analysis. The criteria for better solutions were better values in the metrics of code correctness, conciseness, and complexity. The null hypothesis can be derived from the alternative hypothesis as that the formal methods group solutions have no differences in the solutions regarding the previously mentioned criteria with respect to the control group solutions due to their use of formal analysis.

Generation of an Experimental Plan The control group was randomly sampled of Systems Analysis majors at Miami University of Ohio. The formal methods group were self-selected. To ensure the statistical equivalence between the abilities of the two groups, the American College Testing (ACT) score were measured. The ACT assessment measures high school students' general educational development. Both groups got an elevator task with a functional specification which they had to solve in teams of two people. The students ought to submit an executable and the source code of their program. Submitting UML diagrams or other specifications were optional. After the results had been submitted, the code correctness was measured with test cases that test functional correctness of the implementation. For the conciseness of the code, the lines of code were measured. Complexity of the code was measured by the number of loops, selection statements, and maximal nesting depth.

7.3 Preparation

The requirements for the elevator task were written in the preparation step. Furthermore, six test cases were created for later measuring of code correctness.

7.4 Execution

The execution was done according to the above described plan. The teams from the control group did not submit any UML diagrams. 13 teams submitted an executable and nine of them submitted the source code. In addition, four of the 13 teams submitted pseudocode of the algorithms. In the formal methods group, three of the six teams submitted UML diagrams. Four of the six teams submitted a formal specification in the form of preconditions, postconditions, and invariants for the elevator system.

7.5 Analysis

After calculating the needed metrics mentioned in the design step, there was no significant difference between the formal methods group and control group in conciseness and complexity of the code. However, the test cases only passed for five out of eleven implementations in the control group, which makes 45.5 percent. For the formal method group all six of six implementations passed the test cases, which results in 100 percent. The statistical difference between 45.5 percent and 100 percent is significant and henceforth the code correctness with the usage of formal methods is higher.

7.6 Dissemination and Decision-making

We did not find any replication of the experiment. The cause for this is probably that the experiment is a long-term experiment and therefore only replicable with high effort. The results of the experiment may be used in education to focus on formal methods in courses.

8 Conclusions

Relying on common wisdom for the appropriation of tools and methods can be faulty. For example, if a tool is set as a standard for all projects in a company because “everyone uses it”, it may be unproductive for some projects. To avoid this, reading about different circumstances when a tool is appropriate should be done. For this purpose, researchers should provide the necessary information, e.g. when tools and methods are appropriate for whom and under which conditions.

Experimenting can be one utility to achieve this goal. The different steps that are needed for an experiment were described in this paper. Special care should be taken during the design with the validity of the experiment. Only with external validity the results can generalize to other circumstances, e.g. other companies. For external validity the realism of the experiment should be as high as possible. After an experiment has taken place, replication can help to verify the results.

References

[1] V. R. Basili. The role of controlled experiments in software engineering research. In *Proceedings of the 2006 International Conference on Empirical Software Engineering Issues: Critical Assessment and Future Directions*, pages 33–37. Springer, 2007.

[2] V. R. Basili, F. Shull, and F. Lanubile. Building Knowledge through Families of Experiments. *IEEE Transactions on Software Engineering*, 25:456–473, 1999.

[3] D. M. Berry and W. F. Tichy. Comments on "Formal Methods Application: An Empirical Tale of Software Development". *IEEE Transactions on Software Engineering*, 29:567–571, June 2003.

[4] S. Greenland, J. M. Robins, and J. Pearl. Confounding and Collapsibility in Causal Inference. *Statistical Science*, 14(1):29–46, 1999.

[5] M. Höst, C. Wohlin, and T. Thelin. Experimental context classification: incentives and experience of subjects. In *Proceedings of the 27th International Conference on Software Engineering*, ICSE ’05, pages 470–478. ACM, 2005.

[6] O. Huber. *Das psychologische Experiment: Eine Einfürung*. Hans Huber Verlag, 2005.

[7] A. Jedlitschka and D. Pfahl. Reporting guidelines for controlled experiments in software engineering. *International Symposium on Empirical Software Engineering*, 0:10–20, 2005.

[8] N. Juristo and A. M. Moreno. *Basics of Software Engineering Experimentation*. Kluwer Academic Publishers, 2001.

[9] R. M. Lindsay and A. S. C. Ehrenberg. The Design of Replicated Studies. *The American Statistician*, 47:217–228, August 1993.

[10] S. L. Pfleeger. Experimental design and analysis in software engineering. *SIGSOFT Software Engineering Notes*, 20:22–26, January 1995.

[11] W. R. Shadish, T. D. Cook, and D. T. Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin Company, second edition, July 2001.

[12] F. Shull, V. Basili, J. Carver, J. C. Maldonado, G. H. Travassos, M. Mendonça, and S. Fabbri. Replicating Software Engineering Experiments: Addressing the Tacit Knowledge Problem. In *Proceedings of the 2002 International Symposium on Empirical Software Engineering*, pages 7–17. IEEE Computer Society, 2002.

[13] D. I. K. Sjøberg, B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanovic, E. F. Koren, and M. Vokác. Conducting Realistic Experiments in Software Engineering. In *In Proceedings 1st International Symposium on Empirical Software Engineering*, pages 17–26. IEEE Computer Society, 2002.

[14] D. I. K. Sjøberg, B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanovic, and M. Vokác. *Challenges and Recommendations When Increasing the Realism of Controlled Software Engineering Experiments*, pages 24–38. Lecture Notes in Computer Science, Volume 2765. Springer, 2003.

[15] D. I. K. Sjøberg, J. E. Hannay, O. Hansen, V. By Kampenes, A. Karahasanovic, N.-K. Liborg, and A. C. Rekdal. A Survey of Controlled Experiments in Software Engineering. *IEEE Transactions on Software Engineering*, 31:733–753, September 2005.

[16] A. E. K. Sobel. Empirical results of a software engineering curriculum incorporating formal methods. In *Proceedings of the 31th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '00, pages 157–161. ACM, 2000.

[17] A. E. K. Sobel and M. R. Clarkson. Formal Methods Application: An Empirical Tale of Software Development. *IEEE Transactions on Software Engineering*, 28(3):308–320, 2002.

[18] A. E. K. Sobel and M. R. Clarkson. Response to "Comments on 'Formal Methods Application: An Empirical Tale of Software Development'". *IEEE Transactions on Software Engineering*, 29:572–575, June 2003.